



Московский государственный университет печати имени Ивана Фёдорова
Кафедра медиасистем и технологий

Анна Юрьевна Филиппович

ФАКСИМИЛЬНЫЕ ШРИФТЫ

Лекции по дисциплине "ИСКУССТВО ШРИФТА"

*Для студентов,
осваивающих образовательные программы
подготовки бакалавров и магистров по направлениям
230400 «Информационные технологии и системы»,
230100 «Информатика и вычислительная техника»*

Москва, 2012

Переиздание старинных книг

С развитием информационных технологий способы хранения информации перешли на принципиально другой уровень. И если раньше таким способом была запись на бумажном носителе, что ограничивало объем и уменьшало надежность такого вида хранения, то сейчас использование информационных технологий позволяет нам значительно увеличить эти два показателя.

Многие книги, созданные в эпоху только зарождения печатного дела требуют более эффективного способа хранения. Конечно, с уверенностью нельзя утверждать, что хранение информации на электронном носителе более надежно, так как такой способ хранения еще слишком «молод». То есть, у нас нет тысячелетнего опыта хранения информации в электронном виде. При этом существуют книги, которым сотни лет и, не смотря на пожары и наводнения, до нас дошли пусть не все, но часть из книг прошлого. Однако, недолговечность материала очевидна.

Кроме этого, важно отметить, что электронный способ хранения обеспечит большую доступность информации. Очевидно, что например не каждый человек может позволить купить себе ту или иную книгу, не каждый может прочитать или посмотреть то или иное издание. «Старые» книги разбросаны по музеям и библиотекам всего мира, поэтому доступ к ним ограничен. В связи с этим можно говорить не только об эффективности хранения информации, а также об ее доступности. Поэтому вопрос переиздания книг прошлого и записи их в электронном виде является сейчас актуальным.

Существует несколько способов переиздания старинных книг. Современные печатные технологии базируются на электронном тексте. И даже если печатным источником является оригинал-макет, представляемый в печатном виде, все равно редакция, корректура, верстка осуществляется на компьютере. Поэтому одним из способов переиздания старинных книг является набор текста с последующей корректурой, редакцией и т.д. Но тут важно подчеркнуть некоторую особенность. Любая книга, не важно когда она создана, имеет свой зрительный образ. Этот зрительный образ отражает представления того времени. Поэтому он является вполне понятным и естественным. Если же мы нарушим этот зрительный образ, то, возможно, будет нарушен баланс между внутренним содержанием предмета и его внешним образом. Именно поэтому при переиздании некоторых старинных книг стараются сохранить их первоначальный образ. И такие переиздания называются «факсимильными». Теперь вполне понятным является то, что факсимильные переиздания в точности воспроизводят оригинал.

В первом случае при создании факсимильного издания формируют изображение каждой страницы книги и печатают его в соответствующем формате. Для этого предварительно сканируют и обрабатывают страницы книги.

Недостатки такого способа переиздания очевидны. Если книга находится в плохом состоянии, то изображения страниц требуют значительной обработки. Зачастую улучшить качество изображения невозможно. Кроме этого объем хранения графического файла изображения страницы значительно превышает объем текстового файла.

И наконец, для осуществления поиска по электронной книге, переизданной вышеуказанным способом необходимо будет фрагментировать изображение, осуществлять его индексирование и т.д. Но это касается печатных книг, если переиздается рукопись, то первый способ единственно правильный.

Другой же способ факсимильного переиздания позволяет исключить все вышеперечисленные недостатки. В этом случае создается оригинальная шрифтовая гарнитура, который был набран текст – это так называемая «Факсимильная гарнитура». Далее осуществляется набор, ввод текста с последующей его редакцией, корректурой, версткой, причем каждая сверстанная страница должна полностью повторять оригинал. К

достоинствам такого переиздания следует отнести возможность его хранения в виде электронного текста.

Особенности разработки факсимильных шрифтов

Факсимильный шрифт может совсем не отличаться от любого другого шрифта. Фактически многие существующие шрифты и целые гарнитуры были созданы на основе ранее существующих шрифтов и гарнитур. Сама эволюция печатного дела обусловила этот факт. Поэтому справедливо утверждать, что многие гарнитуры являются в том числе точной копией своих старинных собратьев, то есть являются факсимильными. Но важным является цель, которую преследуют создавая ту или иную гарнитуру. И если цель – факсимильное переиздание, и гарнитуру создают для этого, то безусловно речь идет о факсимильной гарнитуре.

Этапы разработки факсимильных шрифтов

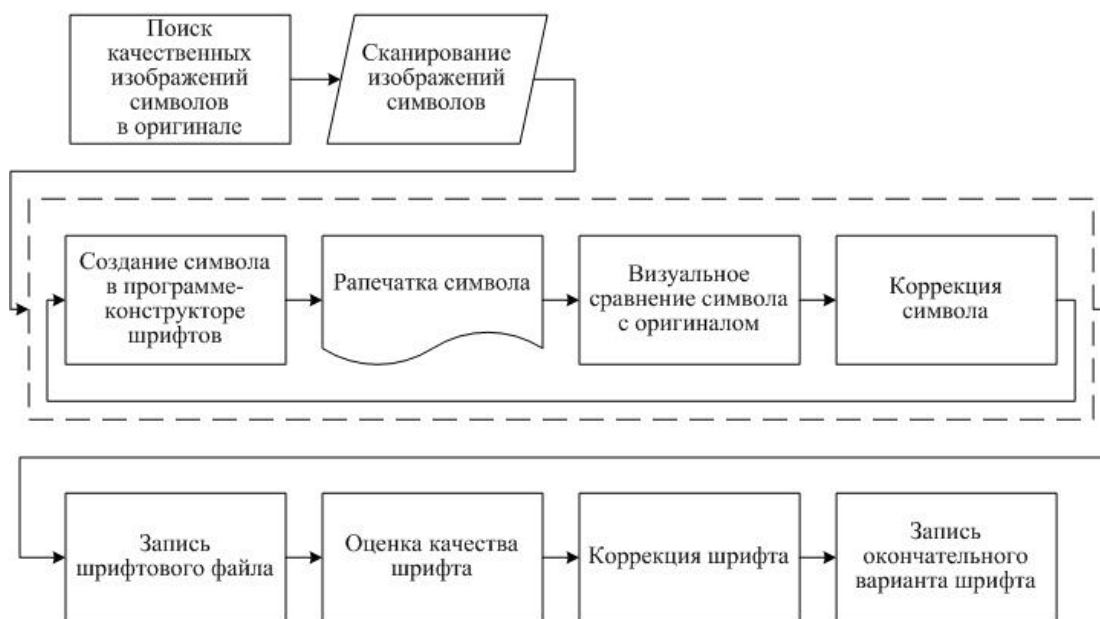


Рис. 1. Схема создания факсимильного шрифта.

Процесс разработки факсимильного шрифта во многом не отличается от «классической схемы» разработки шрифта, однако некоторые существенные отличия существуют. И эти отличия продиктованы необходимостью точного повторения оригинала.

На первом этапе разработки в книге-источнике находятся все буквы алфавита создаваемого шрифта. Причем выбираются наиболее качественные изображения символов, так как возможно оригинал книги находится в плохом состоянии.

Особенностью создания факсимильного символа является возможность использования в качестве прообраза его отсканированного изображения. Поэтому следующим этапом все найденные символы сканируются с высоким разрешением и обрабатываются с помощью растрового редактора, например Adobe Photoshop.

Далее рисуются соответствующие символы в программе-конструкторе шрифтов, к примеру Fontographer. Следует отметить, что использование для этой цели других векторных редакторов, допустим Adobe Illustrator, нежелательно, так как они не содержат в себе всех функций необходимых для профессиональной разработки шрифтов.

На самом деле теоретически, можно создавать не векторный шрифт, а растровый, так как размер букв фиксируем и кроме этого известны параметры всех устройств вывода. То есть если разрешение принтера, на котором будет выводиться оригинал-макет известно и все его характеристики тоже, то можно использовать этот вариант. Однако этот вариант

трудно реализуем. Дело в том, что растровые шрифты сейчас используются для отображения на экране, где разрешение значительно ниже, чем при печати. Разрешение экрана – 72 pixel/inch, а разрешение у стандартного лазерного принтера 600-1200 dpi. Поэтому для вывода на печать используются векторные шрифты. Наиболее распространены такие форматы как TrueType и PostScript. И этими стандартами оперируют все текстовые редакторы, они ориентированы на периферийное оборудование. Вообще такая постановка нерациональна по причине большого объема таких шрифтовых файлов.

Когда создан первый вариант рисуемого символа, тогда осуществляется проверка его на соответствие оригиналу. Для этого создается шрифтовой файл, и символ выводится на печать. Естественно, если оригинал макет будет сдаваться в печатном виде, то символ печатается на том принтере, на котором будет и сам оригинал макет.

Итак, созданный символ печатается на принтере и визуально сравнивается с оригиналом. Далее рисунок символа исправляется до тех пор, пока не будет достигнуто полного совпадения. И так с каждым создаваемым символом. При этом нельзя не учитывать некоторые особенности вывода текста. В зависимости от того, какое устройство вы используете, качество напечатанного символа будет разным. Так к примеру при типографском выводе краска немного расплывается, поэтому буквы становятся немного толще. И сравнивая с буквой, напечатанной на лазерном принтере, стоит штрихи символа сделать немного тоньше, тогда при печати в типографии в результате получится именно так как в оригинале.

У разработчика шрифта может возникнуть вопрос: «Какой символ стоит рисовать первым?» Вопрос конечно немаловажный, с чего начать? Можно конечно начать с первой буквы алфавита и так далее – до последней. Однако, это не очень рационально. Начать лучше с того символа, который содержит в себе элементы других символов. Так, к примеру, буквы: «Н», «К», «И» все имеют одинаковую форму вертикального (основного) штриха, поэтому можно начать с этих букв, так как потом необходимо будет использовать эти элементы в других символах, чтобы достичь единства рисунка. Лучше начать с символов с более простыми рисунками. Кроме этого можно начать со строчных букв, т.к. в тексте строчные буквы используются гораздо чаще прописных, поэтому то, как будет выглядеть текст в будущем, становится более понятным уже на первых этапах разработки шрифта.

Оценка качества шрифта

Чтобы оценить качество шрифта необходимо, во-первых, определить его характеристики. Факсимильный шрифт должен обладать всеми теми характеристиками, которыми должен обладать любой другой шрифт: художественные достоинства, удобочитаемость, емкость.

Кроме вышеперечисленных характеристик шрифта существуют внутренние критерии качества шрифта:

Качество разметки

Правильная разметка отдельных символов и шрифта в целом оказывает очень большое влияние на качество воспроизведения текста, особенно на выводных устройствах с невысокой разрешающей способностью – дисплеях и матричных принтерах.

Полнота набора знаков

Для полноценного использования шрифта необходимо, чтобы он имел полный набор знаков в соответствии с некоторым стандартом. Если шрифт предполагается использовать в одной из программ, работающих под управлением MS Windows, то он должен быть выполнен в соответствии со стандартом 1251 фирмы Microsoft. Минимальный набор знаков, без которого работа со шрифтом будет весьма затруднена, включает все буквы (прописные и строчные), цифры, знаки препинания и некоторые специальные символы,

например тире (его не стоит путать со знаками минус и дефис – это три разных символа!), символ номера или параграфа. Некоторые шрифты имеют только символы русского алфавита. Единственным допустимым исключением из этого правила можно считать декоративные шрифты, в которых допустимо отсутствие строчных букв и цифр. Такие шрифты обычно применяются для выполнения акцентирующих надписей, состоящих всего из нескольких слов, так что полнота набора знаков для них не имеет особого значения.

Правильность кодировки

Шрифт должен не только включать в себя все необходимые знаки. Важно также, чтобы все символы располагались строго на местах, определенным стандартом.

Правильность оформления заголовка

Пожалуй, самые неприятные ошибки в шрифтах связаны с неправильным оформлением заголовка. В самом деле, шрифт вроде бы всем хорош, но пользоваться им нельзя, поскольку ни одна программа его не воспринимает.

Наиболее опасным является неправильное указание уникального идентификатора в Type 1-шрифтах (PostScript). Если два Type 1 -шрифта имеют одинаковое значение этого параметра и используются одновременно (например, в программе АТМ), то растеризатор, как правило, полностью выходит из строя и требует перезагрузки. Во избежание подобных сбоев фирма Adobe производит регистрацию всех производителей шрифтов и присваивает им некоторый диапазон номеров.

Другая группа проблем связана с присвоением шрифтам неправильных имен. В результате шрифты могут неправильно регистрироваться растеризаторами и их использование окажется весьма затрудненным.

Некоторые трудности возникают в случае неправильного указания значений вертикальных размеров шрифта. Например, отсутствие информации о линиях прописных букв, строчных букв, верхних и нижних выносных элементов может привести к неправильному определению кегля шрифта при наборе текста. При этом одинаково заданный кегль для похожих шрифтов будет приводить к совершенно разным результатам.

Соответствие требованиям формата

Наиболее неприятные ошибки возникают в том случае, если шрифт «совсем немного» не соответствует требованиям формата. Разница может заключаться буквально в одном бите шрифтового файла, но рано или поздно она скажется и возникнет ошибка. Такие ошибки трудно обнаружить, они возникают неожиданно и не всегда повторяются даже в случае точного копирования ситуации, в которой ошибка проявилась впервые. Наиболее надежным экспортом TrueType-шрифтов обладает программа Fontographer, а PostScript Type 1-шрифтов – программы FontLab (с точки зрения соответствия формату, Fontographer экспортирует Type 1 шрифты абсолютно корректно, но, в отличие от FontLab, он не поддерживает некоторые тонкости, что заметно ухудшает качество растеризации).

Полнота описания метрических параметров

Как ни странно, но правильность определения метрических параметров шрифта (полей и ширины символов, кернинга и трекинга) оказывает на качество передачи текста большее влияние, чем качество прорисовки отдельных символов. Хорошо проработанный шрифт должен восприниматься гармонично, буквы не должны выпадать или налезать друг на друга.

Относительно поддержки кернинга рекомендуется отдавать предпочтение шрифтам, содержащим достаточно полную таблицу пар кернинга. Поддержка трекинга не имеет особого значения, так как пока только очень немногие программы могут использовать информацию о трекинге, содержащуюся в шрифте. Большинство издательских систем имеют встроенные редакторы трекинга, но, как это ни странно, не воспринимают трекинг,

определенный в шрифтах. Тем не менее, наличие правильно описанного трекинга можно считать полезным.

Качество контуров

Признаками плохого качества контуров являются:

Нарушение гладкости в местах соединения графических примитивов.

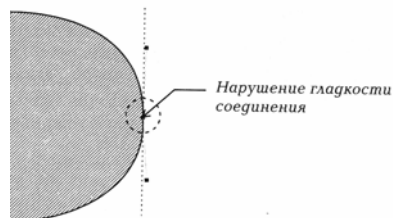


Рис. 2. Пример нарушения гладкости соединения графических примитивов.

Данная ошибка может быть незаметна, однако сильно усложняет работу растеризатора.

Отсутствие выделенных точек экстремумов.

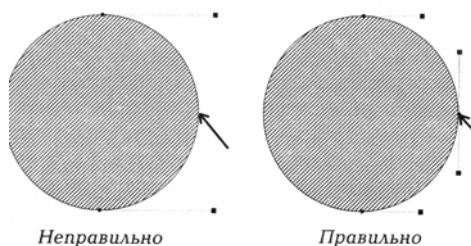


Рис. 3. Отсутствие выделенных точек экстремумов в округлых элементах.

Для нормальной работы растеризатора необходимо, чтобы все экстремумы контуров были выделены в качестве крайних точек кривых или векторов. В случае невыполнения этого правила растеризатор не может автоматически корректировать форму округлых элементов, из-за этого качество разметки резко уменьшается.

Наличие острых внутренних углов.



Рис. 4. Наличие острых углов.

При описании острых углов (меньше 20°) для нормальной работы растеризатора необходимо включать короткий (1-3 единицы) вектор между примитивами, образующими угол. В противном случае форма контура вблизи примитива может сильно измениться.

Использование длинных кривых.



Рис. 5. Примеры некорректных кривых.

При описании сложных элементов не рекомендуется использовать слишком длинные кривые. Причем некоторые кривые, имеющие две точки перегиба или слишком разные длины контрольных векторов, могут нарушить работу растеризатора.

При этом слишком большое количество кривых (большое количество точек, для описания формы кривой) тоже является нежелательным. Необходимо найти «золотую середину» между этими требованиями.

Нарушение вертикальности и горизонтальности штрихов.

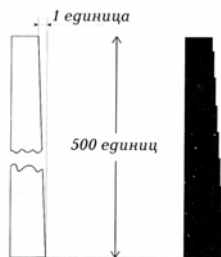


Рис. 6. Нарушение вертикальности штрихов элементов символа.

Когда вектор, образующий вертикальный или горизонтальный штрих немного отстоит от строго вертикального или горизонтального направления, в некоторых случаях это может привести к появлению зубцов на линиях символа и изменить толщину штриха.

Нарушение размеров символов.

При смещении символов относительно базовой линии, разных размеров символов или различиях в величинах наплывов у округлых символов визуально нарушаются размеры символов, и в результате при выводе строки текста возникает лесенка внизу или сверху строки.

Особенности использования сканированных изображений

Для использования в Fontographer необходимо, чтобы отсканированные изображения были в битовом формате (Bitmap). Для хранения каждого пикселя битового изображения используется 1 бит. То есть все пиксели будут окрашены либо в белый, либо в черный цвет. Чтобы добавить изображение необходимо скопировать его в буфер обмена в любом графическом редакторе и вставить в окно символа. При этом в Fontographer есть слой, предназначенный для таких изображений: Template. Важно отметить, что изображение в этом слое показывается с экраным разрешением, но, не смотря на это, качество исходного изображения и его размер является важным. Поэтому сканировать изображение необходимо с высоким разрешением не менее 300 dpi. Тогда при автоматической обводке в Fontographer результат будет достаточно точным.

Если исходное изображение было представлено в градациях серого (Grayscale), то Fontographer автоматически преобразует его в Bitmap. Но для достижения наилучшего результата желательно предварительно преобразовать его в Bitmap. Для этого можно использовать функции Adobe Photoshop.

Grayscale изображения еще могут называть 8 битовыми, так как пиксели могут иметь 256 оттенков серого цвета. Пиксели характеризуются величинами яркости в диапазоне от 0 (яркость отсутствует, черный цвет) до 255 (полная яркость, белый цвет). Когда Grayscale изображение преобразуется в битовое, необходимо решить, какие пиксели будут закрасены в черный, а какие в белый цвет.

В Adobe Photoshop существует несколько методов, которые позволяют преобразовать изображение в Bitmap. Рассмотрим подробно каждый из них.

Метод *50% Threshold* (Порог 50%) устанавливает границу яркости в диапазоне от 0 до 255 ровно посередине – 128. Все пиксели справа от границы окрашиваются в белый, а слева – в черный цвет. То есть все тона, содержащие более 50% черного цвета, становятся

черными, а все тона, которые содержат менее 50% черного цвета, становятся белыми [О'Куин, 1998]. Результаты использования данного метода представлены на рис.7.

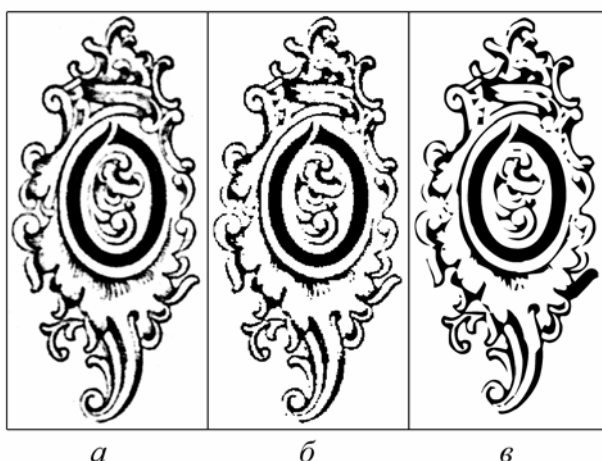


Рис.7. Использование метода 50% Threshold: а – исходное изображение, режим: Grayscale; б – использование метода; в – результат автоматической обводки.

Именно этот же метод используется в Fontographer. Однако существует возможность точно задавать значение порога. Для этого нужно воспользоваться командой: Image (Изображение) → Adjust (Коррекция) → Threshold (Изогелия). При этом появится окно с гистограммой изображения (рис. 8). Передвигая бегунок, можно изменять значение порога, при этом будет видно как изменяется изображение.

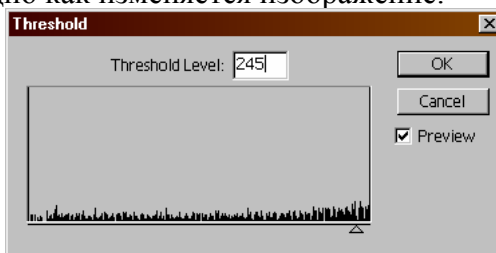


Рис. 8. Окно настройки порога: Threshold.

Результаты использования метода Threshold (Изогелия) представлены на рис. 9.

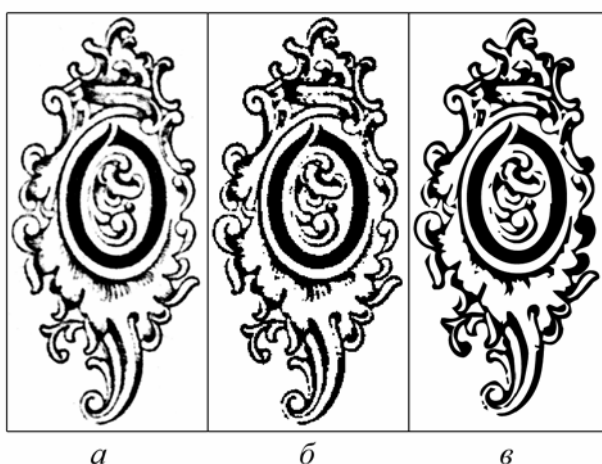


Рис. 9. Использование метода Threshold с точным значением порога: а – исходное изображение, режим: Grayscale; б – использование метода (граница – 200); в – результат автоматической обводки.

Метод *Pattern Dither* (Регулярное распределение) «разбрасывает» пиксели, чтобы имитировать в изображении градации серого. При этом Photoshop использует заранее

определенные геометрические шаблоны. Результаты использования данного метода представлены на рис. 10.

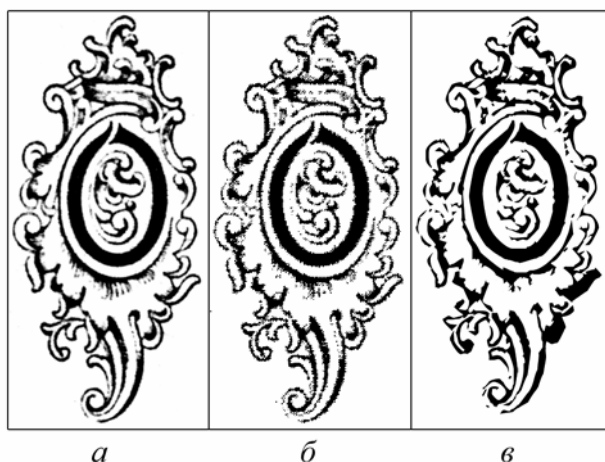


Рис.10. Использование метода Pattern Dither: а – исходное изображение, режим: Grayscale; б – использование метода; в – результат автоматической обводки.

Метод *Diffusion Dither* (Случайное распределение) аналогичен Pattern Dither, но при этом пиксели распределяются случайным образом. Это метод неэффективен для разрешения, превышающего 300 dpi. Результаты использования данного метода представлены в рис. 11.

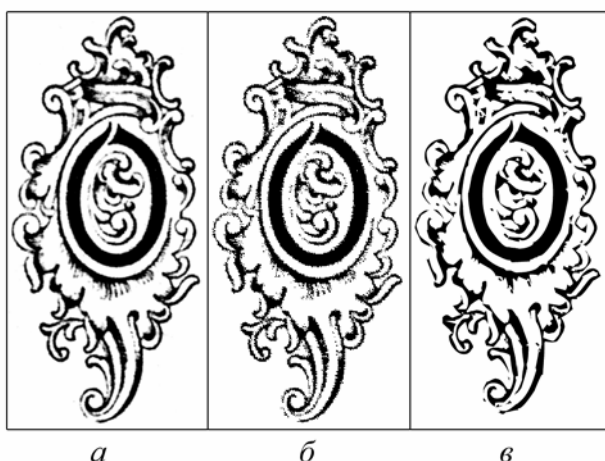


Рис. 11. Использование метода Diffusion Dither: а – исходное изображение, режим: Grayscale; б – использование метода; в – результат автоматической обводки.

Метод *Halftone Screen* (Полутоновой растр) позволяет превращать тона серого в наборы полутонных точек. При этом можно задать следующие установки:

- *Frequency* (*Линиатура*). Эта величина определяет размер смоделированных полутонных точек.
- *Angle* (*Угол*). Эта опция позволяет задать угол наклона растра или угол поворота полутонной сетки.
- *Shape* (*Форма*). Эта опция позволяет выбрать форму полутонной точки (круг, ромб, эллипс, линия, квадрат, крест).

Полутонной растр может быть использован только для создания специального эффекта. Результаты использования данного метода представлены на рис. 12, в данном примере использовались следующие установки: Frequency = 53 lines/inch; Angle = 45 degrees; Shape = round.

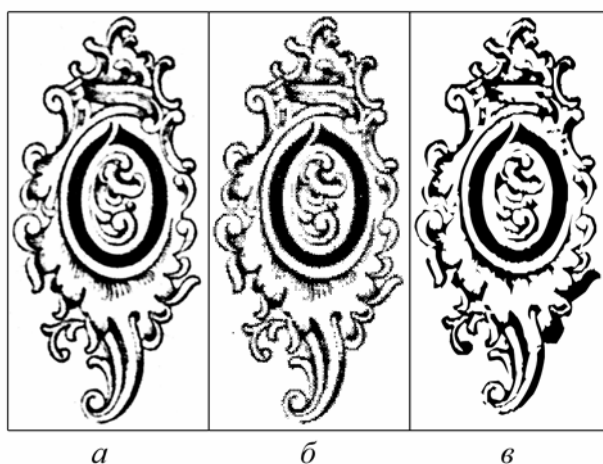


Рис. 12. Использование метода Halftone Screen: а – исходное изображение, режим: Grayscale; б – использование метода; в – результат автоматической обводки.

Метод *Custom Pattern* (Заказной растр) позволяет использовать образец, заранее заданный с помощью команды: Edit (Редактирование) → Define Pattern (Определить образец). Как и в случае применения метода 50% Threshold, цвета изображения преобразуются в черный и белый, но повторяющийся образец появляется только в тех областях, окрашенных в цвета, содержащие более 50% черного цвета.

Таким образом, для обработки изображений шрифтовых элементов или орнаментов наиболее подходит метод 50% Threshold или Threshold. Эти методы позволяют получать ровно окрашенные контрастные битовые изображения. Тем более, как показал опыт, при автоматической обводке в Fontographer таких изображений линии получаются более гладкими. Если исходное изображение слишком светлое и некоторые детали не видны, то можно отредактировать битовое изображение, изменяя порог с помощью метода Threshold, тем самым, увеличивая количество черных точек, то есть детальность и толщину линий.

Недостатки качества контуров при использовании функции автоматической обводки в Fontographer


При использовании функций автоматической обводки в Fontographer возникают недостатки связанные с качеством контуров. Данные ошибки могут плохо влиять на результат разметки символа, более того при определенных ошибках может быть нарушен вид символа настолько, что он станет нечитаемым. Т.о. при построении контуров стоит обратить внимание качество контуров. Основные признаки плохого качества контуров в Fontographer следующие:

Наличие незамкнутых контуров.

Для правильного отображения шрифтового знака необходимо, чтобы все контуры были замкнуты.



Рис. 13. Наличие незамкнутых контуров: а – неправильно, б – правильно.

Разрыв контура в Fontographer обозначается: . Аналогично обозначается начало контура, из-за этого может возникнуть путаница. Т.о. в замкнутом контуре может быть только одна точка с таким обозначением, иногда при неправильном отображении символа необходимо проверять начальные точки на разрыв (рис. 13).

Наличие отдельных точек.

Подобно тому, как плохо сказывается наличие разрывов при отображении символов, и их разметки наличие отдельных точек является недопустимым (рис. 14). Точки используются для описания контуров, поэтому отдельные точки могут трактоваться программой как часть контура, искажая и изменяя его форму.

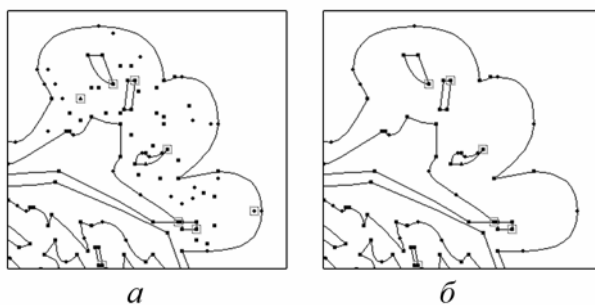



Рис. 14. Наличие отдельных точек: а – неправильно, б – правильно.

Наложение точек.

При наложении нескольких точек в Fontographer появляется следующее обозначение: . При заполнении символа это никак не сказывается, однако эти точки являются лишними и ухудшают качество разметки (рис. 15).

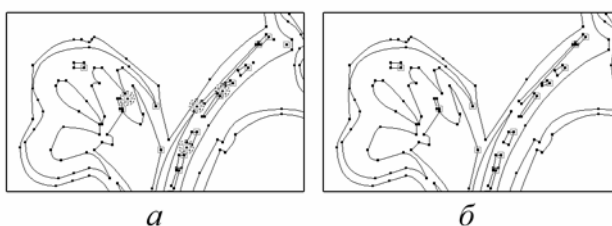


Рис. 15. Наложение точек: а – неправильно, б – правильно.

Ошибки в заполнении контуров.

Ошибки в заполнении контуров могут быть связаны с наличием разрывов или несоответствии типов внешних и внутренних контуров. Так внутренний контур является внешним и при заполнении шрифтовой символ отображается некорректно (рис. 16).

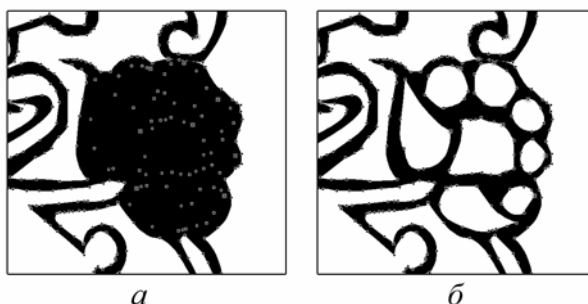


Рис.16. Ошибки в заполнении контуров шрифтовых знаков: а – неправильно, б – правильно.

Избыточность в описании контура (слишком много точек).

Такая ошибка появляется, если при трассировке растрового изображения в настройках указать слишком высокое значение точности обводки. Количество точек,

которые требуются для описания контуров должно быть оптимальным (рис.17). Каких-то точных показателей не существует, это зависит от самого контура.

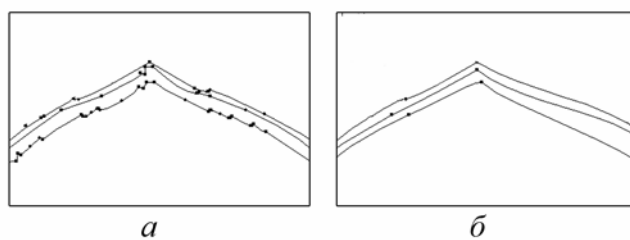


Рис. 17. Избыточность в описании контуров: а – неправильно, б – правильно.

Пересечение контуров.

Пересечение контуров является недопустимым. Это может быть связано с наличием слишком длинных кривых, петель или с наложением контуров (рис. 18).

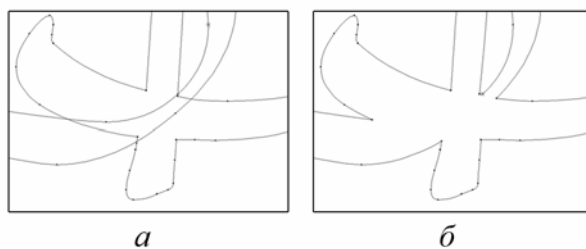


Рис. 18. Пересечение контуров: а – неправильно, б – правильно.